

# GitHub Actions × Git Flow でHTML一覧 (index.html) を自動更新する手順

## 目的

- HTMLファイルをどんどん追加しても、**index.html**を手で編集しない
- `git push` だけで、**GitHub Actions** が自動で一覧を更新
- GitHub Pages で **スマホから手軽に閲覧**できる状態を保つ

## 全体像 (Git Flow)

```
ローカルでHTML作成
↓
featureブランチでcommit
↓
mainへmerge (push)
↓
GitHub Actions起動
↓
Python実行 (index.html生成)
↓
自動commit & push
↓
GitHub Pages反映
```

## ディレクトリ構成 (前提)

```
/
├── index.html          ← トップ入口 (固定・触らない)
├── pages/
│   ├── index.html      ← ★自動生成される一覧
│   ├── 2025-03-20_setup.html
│   ├── 2025-03-22_memo.html
│   └── ...
├── tools/
│   └── build_index.py   ← 一覧生成スクリプト
└── .github/
    └── workflows/
        └── build-index.yml ← GitHub Actions
```

## 命名ルール（重要）

YYYY-MM-DD\_タイトル.html

- 日付順で自動ソート可能
- 人間にも機械にも分かりやすい
- index生成ロジックが単純になる

## Step 1：HTMLを追加する（人がやる作業）

pages/2025-03-25\_trouble.html

- 内容は自由
- リンク管理は一切不要

## Step 2：一覧生成スクリプト（Python）

tools/build\_index.py

```
from pathlib import Path

pages_dir = Path("pages")
html_files = sorted(
    [f for f in pages_dir.glob("*.html") if f.name != "index.html"],
    reverse=True
)

lines = [
    "<!doctype html>",
    "<html><head>",
    "<meta charset='utf-8'>",
    "<meta name='viewport' content='width=device-width, initial-scale=1'>",
    "<title>資料一覧</title>",
    "</head><body>",
    "<h1>📄 資料一覧</h1>",
    "<ul>"
]

for f in html_files:
    title = f.stem.replace("_", " ")
    lines.append(f"<li><a href='{f.name}'>{title}</a></li>")

lines += [
    "</ul>",
    "</body></html>"
]
```

```
]
```

```
(pages_dir / "index.html").write_text("\n".join(lines), encoding="utf-8")
```

## Step 3 : GitHub Actions を設定する

`.github/workflows/build-index.yml`

```
name: Build pages index

on:
  push:
    paths:
      - "pages/*.html"

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Set up Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.x"


      - name: Build index.html
        run: python tools/build_index.py

      - name: Commit and push if changed
        run: |
          git config user.name "github-actions"
          git config user.email "github-actions@github.com"
          git add pages/index.html
          git diff --cached --quiet || git commit -m "auto: update pages index"
          git push
```

## Step 4 : 日常運用（これだけ）

```
git checkout -b feature/add-doc
git add pages/2025-03-25_trouble.html
git commit -m "add trouble doc"
git checkout main
```

```
git merge feature/add-doc  
git push
```

 index.html は 自動更新

## スマホでの閲覧導線

GitHubアプリ  
→ README  
→ <https://ユーザー名.github.io/リポジトリ名/pages/>  
→ 一覧表示

## ハマりどころチェック

- [] GitHub Pages が main / root で有効か
- [] pages/index.html は手動で編集していないか
- [] Actions が push 時に起動しているか
- [] 無限ループしていないか (paths指定あり)

## この設計の意図（なぜこうしているか）

- 人がやるべきなのは「内容を書く」だけ
- 構造・並び・漏れは 機械に任せる
- Git Flow と CI を 個人ナレッジ運用に転用することで 作業コストを増やさず、長く続けられる

## 次の拡張案（必要になったら）

- `<title>` タグから表示名を取得
- 年月フォルダ対応
- README.md も自動更新
- GitHub Actions 成功通知

これで、「HTMLを追加 → push →勝手に一覧更新 → スマホで即閲覧」が完成です。